CATRAC: Context-Aware Trust- and Role-Based Access Control for Composite Web Services

Cesar Ghali  Ali Chehab  Ayman Kayssi

Electrical and Computer Engineering Department
American University of Beirut
Beirut 1107 2020, Lebanon
{csg04, chehab, ayman}@aub.edu.lb

Abstract—Web Services are at the heart of many Internet-based e-business systems. Security issues in web services are critical for the continuity of the provided services. Solutions such as Role-Based Access Control and Trust-Based Access Control were proposed to address threats to security in single Web Service scenarios. These solutions do not fully provide the required security level in situations related to composite Web Services. We present a new security framework related to composite Web Services and that combines role-based and trust-based access control. We verify the correctness and performance of the proposed framework and show simulation results from a prototype implementation.

Keywords—web services, access control, roles, trust, security.

I. INTRODUCTION

Web Services are a set of methods and functions that are described by a Web Service Description Language (WSDL) and published using Universal Description Discovery and Integration (UDDI) [1]. Web Services are at the heart of many e-business systems. Thus, securing the Web Service is critical, with client access control being the most widely used technique. Role-Based Access Control (RBAC) assigns roles to clients in order to access a particular Web Service. Trust-Based Access Control (TBAC) takes into account the client’s history of activities and behaviors to provide a trust level that can be used to differentiate between trusted and non-trusted clients. Other access control schemes have also been proposed, such as context-aware schemes. Securing the network communication between clients and Web Service providers is also needed, and is typically achieved using encryption. Public-key [2] and symmetric key [3] techniques are used to protect sensitive information such as client credentials, passwords, transaction data, etc.

The rest of the paper is organized as follows: Section II discusses previous works related to Web Service security. Section III presents the CATRAC system design and architecture. The protocol used in the communication between CATRAC entities is presented in Section IV. In Section V we analyze the security of CATRAC. In Section VI we present the proposed trust management mechanism. Section VII describes the simulation performance and results obtained when testing CATRAC. Conclusions are presented in Section VIII.

II. PREVIOUS WORK

One of the most important and widely used Web Service access control schemes is Role-Based Access Control (RBAC) [4-7]. In such access control schemes, clients are assigned roles that contain permissions in order to gain a secure access to specific Web Services. Although this concept is relatively simple in single Web Service scenarios, handing over roles to other services or providers in a composite Web Service context is a challenge. In single service situations, the client’s role is checked by the invoked service which then allows or denies the request based on the permissions that the role carries. However, in composite Web Services, a service might invoke other services. The problem here lies in accessing the other services using the original client’s permissions, rather than using the intermediate service permissions.

The Global Roles scheme [4] is one technique where global Web Services rely on global roles. This composition combines more than one local service from different providers. Therefore, the global roles must contain information about all the local services invoked by the global service. Other systems [8] use policy files, instead of global roles, in both single and composite scenarios in order to check the validity of the client’s request and its possession of the right permissions. Further analysis and operations must be performed in composite Web Services to combine the policy files for all involved services.

Attribute-Based Access Control (ABAC) models add more dynamicity to the traditional RBAC systems [9]. These new models make use of attributes owned by the clients, the providers, and some other attributes related to the environment. Decisions are be made to allow or deny the request based on all these attributes.

Trust-Based Access Control systems [10-12] are different from the previous access control schemes since clients’ trust levels are dynamically calculated based on some statistical analysis of behaviors, activities and previous access attempts. Thus, service violations and bad client behavior lead to a decrease of the trust level, whereas good behavior leads to an increase in the trust level.

Moreover, Ray et al have proposed a trust model based not on a binary trust concept of trust or distrust a specific entity, but on a vector containing trust values [13]. They calculated
the trust level of the client to be fit in the range of \([-1,+1]\) where the positive values express trust level and the negative values express distrust level. The neutral trust level of a client is equal to zero. Furthermore, Ray et al proposed, in [14], a new model for assessing trust relationship based on previous clients access attempts and behavior in a certain or related context. However, such systems are not able to assess the trust value in case of totally new entities, or known entities but in new context.

The system proposed in this paper, CATRAC, is based on a combination of context-aware, role- and trust-based access controls. Hence, three conditions must be met in order to gain access to a specific Web Service: client attributes are authenticated by the Web Service provider, client’s global role must be valid and contains the right permissions, and the client’s trust level should be at least equal to the minimum required level for the particular service.

The roles that are assigned to clients are issued, signed and verified by a trusted third party called the Role Authority. Trust levels are not expressed as binary representation of trusted or distrusted entities, but as vector of values range from 0 to 10, where 10 indicates a fully trusted client and 0 implies a fully distrusted clients. In addition, new clients are handled by assigning them the neutral or the uncertainty trust level of 5.

III. CATRAC SYSTEM DESIGN

In order to fully present the CATRAC system in handling composite Web Services, we will start with the architecture of a single Web Service and then expand it to cover the composite Web Service scenario.

A. CATRAC Core

Possession of roles by clients allows them to carry permissions to access certain services and associated resources. The roles expire after a certain time based on the sensitivity of the data exposed or actions executed by the service. On the other hand, roles are not used alone, as a trust level is assigned to clients based on past behavior.

The same applies in the case of composite Web Services. CATRAC system is designed to provide clients with transparent access to composite Web Services. Clients do not know if the provider is single or composite. From their perspective, it is always a single Web Service scenario, while in reality it might be a composite one. In composite scenarios, clients’ trust levels are calculated in the intermediate provider each time clients access a specific Web Service. When a new client registers at an intermediate provider, the latter contacts the single providers and requests a reputation of this client, if any. These reputation values are weighted and used to calculate an initial trust level for the new client.

As a result, role possession by clients and trust level verification by providers fully ensure that only the authenticated and authorized clients can access the system.

B. CATRAC Architecture

CATRAC is a composite Web Service access control system that consists of three entities:

- **Role Authority**: The Role Authority (RA) is responsible of creating and digitally signing global roles upon the request of the providers on behalf of the client. After the creation of each role, the RA digitally signs it in order to prevent any tempering of its contents.
  
  In addition to global role creation and signing, the RA manages the client registration. When a new client registers with a specific provider, the provider issues a request to the RA to check if it is a new client to the whole CATRAC system or if it is already registered with other providers. This is helpful in order to obtain the reputation discussed in Section III-A.
  
  Moreover, the authority plays a role in single and composite Web Service scenarios. In both cases clients are issued a role from the RA, either by the request of the single or intermediate provider. This whole process is performed in the background without the clients’ knowledge or involvement.
  
- **Providers**: These entities have three major roles: offering Web Services to end clients, requesting roles from the RA on behalf of the clients, and verifying clients trust levels before accessing any resources.
  
- **Clients**: Clients want to gain access to Web Services from a specific provider. They are first assigned roles by the providers and then, their access attempts are recorded in the provider’s database in order to make an estimation of their trust levels the next time they access the same resources.

In the case of composite Web Services, these three entities still exist. What is different in this case is the way they are connected together. For instance, Imazon is a provider that might be connected to two other providers, ABM and Cell. When a client requests data that Imazon does not have, Imazon obtains it from ABM and Cell. Although Imazon is an intermediate provider, it contacts the RA to issue roles for its clients, in a similar way a single provider does.
IV. CATRAC ENTITIES COMMUNICATION PROTOCOLS

In order for CATRAC to work properly and securely, entities in the system must have appropriate and secure communication protocols to exchange data during service requests and responses. The six communication steps are: 1- registration, 2- verification, 3- activation, 4- login, 5- request global role, and 6- request service. Figure 2 demonstrates these steps and their execution sequence.

![Figure 2. CATRAC Protocols in Sequence](image)

A. Registration Protocol

The main purpose of this protocol is to register new clients at a specific provider. The RA is involved in this protocol since the provider checks the existence of the new client with the help of the authority. A UniqueID is used to identify and distinguish among clients. During the registration step, the clients must provide some required attributes they are supposed to possess, such as registration numbers, certificates, etc. from their organizations. This information will be used later on while creating the role.

B. Verifying the Client Attribute

This is an offline procedure performed by the RA. This step is application specific. As an example, checking if the client is a student at a specific university needs a confirmation from the university registrar office.

Once the attributes of the client are verified, the client is contacted by the RA. As an example, an email containing an activation link may be sent by the RA to the client once the identity verification step is completed.

C. Activation Protocol

In this protocol, clients activate their pre-registered accounts. This is a mandatory step before clients can request Web Services.

D. Login Protocol

This protocol is used to verify the clients’ credentials during the login process.

E. Request Global Role Protocol

This protocol is invoked when logged in clients try to access a Web Service without the possession of a role. This protocol involves the three entities of the CATRAC system: A client issues a request from the provider which relays it to the RA. The authority creates and signs a new global role for this client. The expiry date of a global role depends on two factors: the sensitivity of the service and the client attributes. For instance, clients with low attribute levels have smaller validity durations. However, the whole process is denied if the client does not have the sufficient attributes to access the requested Web Service.

In addition, the provider encrypts the new role using a symmetric key shared with the client, and sends the result back to the client to be saved locally.

F. Request Service Protocol

After a global role is assigned to the client, it can be combined with the request of the service. The provider first passes the received global role to the RA to check its format, signature and expiration date. The RA sends back the result, and accordingly the provider acts by either invoking the service and sending back the results, or by rejecting the whole request. In addition, the results of executing the Web Service may be transmitted in an encrypted fashion using a session key generated by the client and sent to the provider with the request.

G. The Six Protocols in Composite Web Service Scenarios

The preceding six communication protocols are also used in the case of composite Web Services. When requesting a service from a composite Web Service, the intermediate provider, such as Imazon in the example above, applies all the steps in the Request Service Protocol except for invoking local services. Instead, it forwards the request to services in other providers, such as ABM or Cell, in order to process it. Later on, the intermediate provider combines the obtained results from the single providers and sends them back to the end client.

V. SECURITY OF THE SYSTEM

CATRAC is secured using the following techniques in order to thwart attacker attempts.

- Encryption: Confidentiality and authentication are achieved using symmetric and asymmetric key cryptography.
- Replay Attack Detection and Prevention: Nonces and timestamps are attached with each request in order to detect replay attacks. The destination verifies the timestamp and the freshness of the nonce.
- Signing of Roles: The digital signature protects and provides integrity for the contents of the role, including its expiration date.
- Man in the Middle: A man in the middle cannot extract any useful information from captured packets since all the transferred data is encrypted.
- Malicious Client: If a malicious client illegally possessed a client role, it would not be able to use its own credentials to access services and resources. The reason is that roles are encrypted using a key derived from the client identity.
- Invalid Packets: The service provider detects and drops invalid packets since their format is first checked before they are processed.

VI. TRUST MANAGEMENT

As mentioned earlier, RBAC systems check if the client has the appropriate role to access a particular Web Service.
However, checking the trustworthiness of the client gives more assurance to the provider. The client’s trust level is checked during every access attempt in order to make sure that it is high enough to access the target Web Service. This trust level is obtained based on the client’s past behavior. In the following, we will show how trust levels are calculated and managed in CATRAC.

A. Dynamic Trust Level

In CATRAC, clients accumulate trust points when their behavior is considered good, otherwise they lose trust points. The trust level is a real value in the range of 0 to 10. However, in the case of a totally new client, it is assigned a neutral trust level equal to 5.

In addition, clients’ trust levels must be decayed to the neutral value gradually with time. The reason is that trusted clients might not remain trustworthy in the future, and vice versa. Decaying the client trust level depends on a time factor that takes into account the date of the client’s last activity. The decayed trust level can be calculated dynamically using one of the following equations. Notice that Equation 1 applies when \( T_L_c \geq T_L_N \), and Equation 2 applies when \( T_L_c < T_L_N \).

\[
D_{T_L_c} = \left( T_L_c - T_L_N \right) \times e^{\frac{-t}{\text{mem}_a} } + T_L_N \quad (1)
\]

\[
D_{T_L_c} = T_L_N \times \left( 1 - e^{\frac{-t}{\text{mem}_a} } \right) \quad (2)
\]

Where:
- \( D_{T_L_c} \) is the decayed trust level for client \( c \).
- \( T_L_c \) is the current trust level for client \( c \) before decaying.
- \( T_L_N \) is the neutral trust level.
- \( t \) is the time elapsed since the client’s last activity.
- \( \text{mem}_a \) is a memory factor (or time constant) determining the duration required for the \( T_L_c \) to decay from the maximum trusted or untrusted level to the neutral level.

Figure 3 shows the operation of the decaying trust level with the time unit used for \( t \) being in days.

VII. SIMULATION AND RESULTS

To implement all the functionality needed by the system from client registration to service request, a CATRAC application-independent framework was developed. On top of this framework, an e-commerce application was implemented to test CATRAC operability and performance in a composite Web Service context.

The e-commerce application consists of three providers, named Imazon, Cell, and ABM, and one RA, as shown in Figure 1. Imazon is a composite Web Service provider that is connected to the providers Cell and ABM. All the three providers are connected to the RA in order to perform all role-related operations. Seven example Web Services are implemented by both ABM and Cell. These services are:

- GetLaptopsList: returns the list containing the existing laptop computers at a specific provider.
- GetLaptopSpecs: returns the specifications of a particular laptop computer.
- GetPrintersList: returns the list containing the existing printers at a specific provider.
- GetPrinterSpecs: returns the specifications of a particular printer.
- OrderLaptop: buy a specific laptop from a specific provider. Invoking this service increases the trust level.
- OrderPrinter: buy a specific printer from a specific provider. Invoking this service increases the trust level.
- CancelOrder: cancel a laptop or a service order. Invoking this service decreases the trust level.

Moreover, Imazon also provides the previous seven Web Services but does not possess the requested information. Therefore, since Imazon is an intermediate provider, it gathers the data from ABM and Cell and sends it back to the client.

The e-commerce application along with the CATRAC framework was tested in a LAN environment consisting of two servers: one implementing the three providers and the other representing the RA. Also, multiple clients were requesting web services from the providers simultaneously in order to test CATRAC’s ability of handling multiple requests simultaneously.

The measured performance figures in the experiments that were performed are related to the required duration to interpret and handle the clients’ requests. The simulation process starts first with 100 clients randomly requesting different services from a provider for 10 minutes. For every single request, the processing duration is logged. This duration is divided into two measurements: time at the provider and time at the role authority. In addition, the duration for processing a role request at the provider and the RA are also recorded.

Based on the logged values, the curves in Figure 4 are sketched. Figures 4 (a) and (c) illustrate the relation between the global role request duration in the provider and the role authority, respectively, versus the number of clients. The curves in Figures 4 (b) and (d) show the relation between the service request duration in the provider and the global role check duration in the role authority, respectively, versus the
number of clients.

It is obvious from the curves that the duration of both the global role request and the service request increases as the number of clients increases. Increasing the number of clients affects the server load and performance, and affects the database engine performance. In all cases, the latency of the system increases quadratically with the number of clients. At a level of 2000 clients, the latency is below the one second threshold.

VIII. CONCLUSIONS

In this paper, we presented CATRAC, a new access control scheme for composite Web Services. CATRAC combines role-based and trust-based access controls in order to achieve a higher level of control and security. Symmetric key and public key cryptography are used for authentication, confidentiality, and integrity of data. CATRAC is applicable in single Web Service scenarios as well as in composite scenarios. The performance of a prototype implementation shows that the system is scalable to thousands of users.

ACKNOWLEDGMENT

The authors would like to thank and acknowledge the support of the Lebanese National Council for Scientific Research and the AUB University Research Board.

REFERENCES